CLAIMS:

What is claimed is:

Sub B3

1. A method of processing a source code statement written in a programming language, the method comprising the computer-implemented steps of:

parsing a document type definition file for a markup language;

parsing a source code statement from a source code file;

selecting an element defined in the document type definition file based on an association between the element and an identifier of a routine in the source code statement; and

writing the selected element to a markup language file.

Sub [illegible]

2. The method of claim 1 wherein the source code statement comprises parameters for the routine and wherein the element comprises an attribute list corresponding to the parameters.

3. The method of claim 2 wherein the selected element written to the markup language file comprises an attribute list of values for the parameters passed to the routine.

4. The method of claim 1 wherein the routine is a procedure, subroutine, function, method, class, or software object.

5. A method of processing a markup language element, the method comprising the computer-implemented steps of:

parsing a document type definition file for the markup language;

parsing a markup language element from a markup language file;

selecting an element defined in the document type definition file that is equivalent to the markup language element from the markup language file;

generating a source code statement using an identifier of a routine within the selected element; and

writing the source code statement to an output file.

6. A method of generating a markup language file, the method comprising the computer-implemented steps of:

executing an application program;

parsing a document type definition file for a markup language;

selecting an element defined in the document type definition file based on a routine called by the application program; and

writing the selected element to a markup language file.

7. The method of claim 6 wherein the element comprises an attribute list corresponding to parameters for the routine.

8. The method of claim 6 wherein the selected element written to the markup language file comprises an

attribute list corresponding to values for the parameters passed to the routine.

9. The method of claim 6 wherein the application program is written in Java programming language.

10. The method of claim 9 wherein the routine is an extended class method.

11. The method of claim 9 wherein the routine is a Graphics class method.

12. A data processing system for processing a source code statement written in a programming language, the data processing system comprising:

first parsing means for parsing a document type definition file for a markup language;

second parsing means for parsing a source code statement from a source code file;

selecting means for selecting an element defined in the document type definition file based on an association between the element and an identifier of a routine in the source code statement; and

writing means for writing the selected element to a markup language file.

13. The data processing system of claim 12 wherein the source code statement comprises parameters for the routine and wherein the element comprises an attribute list corresponding to the parameters.

14. The data processing system of claim 13 wherein the selected element written to the markup language file comprises an attribute list of values for the parameters passed to the routine.

15. The data processing system of claim 12 wherein the routine is a procedure, subroutine, function, method, class, or software object.

16. A data processing system for processing a markup language element, the data processing system comprising:

first parsing means for parsing a document type definition file for the markup language;

second parsing means for parsing a markup language element from a markup language file;

selecting means for selecting an element defined in the document type definition file that is equivalent to the markup language element from the markup language file;

generating means for generating a source code statement using an identifier of a routine within the selected element; and

writing means for writing the source code statement to an output file.

17. A data processing system for generating a markup language file, the data processing system comprising:

executing means for executing an application program;

parsing means for parsing a document type definition file for a markup language;

selecting means for selecting an element defined in the document type definition file based on a routine called by the application program; and

writing means for writing the selected element to a markup language file.

18. The data processing system of claim 17 wherein the element comprises an attribute list of parameters for the routine.

19. The data processing system of claim 17 wherein the selected element written to the markup language file comprises an attribute list of values for the parameters passed to the routine.

20. The data processing system of claim 17 wherein the application program is written in Java programming language.

21. The data processing system of claim 20 wherein the routine is an extended class method.

22. The data processing system of claim 20 wherein the routine is a Graphics class method.

23. A computer program product in a computer readable medium for use in a data processing system for processing a source code statement written in a programming language, the computer program product comprising:

first instructions for parsing a document type definition file for a markup language;

second instructions for parsing a source code statement from a source code file;

third instructions for selecting an element defined in the document type definition file based on an association between the element and an identifier of a routine in the source code statement; and

fourth instructions for writing the selected element to a markup language file.

24. A computer program product on a computer readable medium for use in a data processing system for processing a markup language element, the computer program product comprising:

first instructions for parsing a document type definition file for the markup language;

second instructions for parsing a markup language element from a markup language file;

third instructions for selecting an element defined in the document type definition file that is equivalent to the markup language element from the markup language file;

fourth instructions for generating a source code statement using an identifier of a routine within the selected element; and

fifth instructions for writing the source code statement to an output file.

25. A computer program product on a computer readable medium for use in a data processing system for processing a markup language file, the computer program product comprising:

first instructions for executing an application program;

second instructions for parsing a document type definition file for a markup language;

third instructions for selecting an element defined in the document type definition file based on a routine called by the application program; and

fourth instructions for writing the selected element to a markup language file.

26. A method of processing a source code statement written in a programming language, the method comprising the computer-implemented steps of:

parsing a grammar input file for a markup language;

parsing a source code statement from a source code file;

selecting a language syntax construct defined in the grammar input file based on an association between the language syntax construct and an identifier of a routine in the source code statement; and

writing the selected language syntax construct to a markup language file.